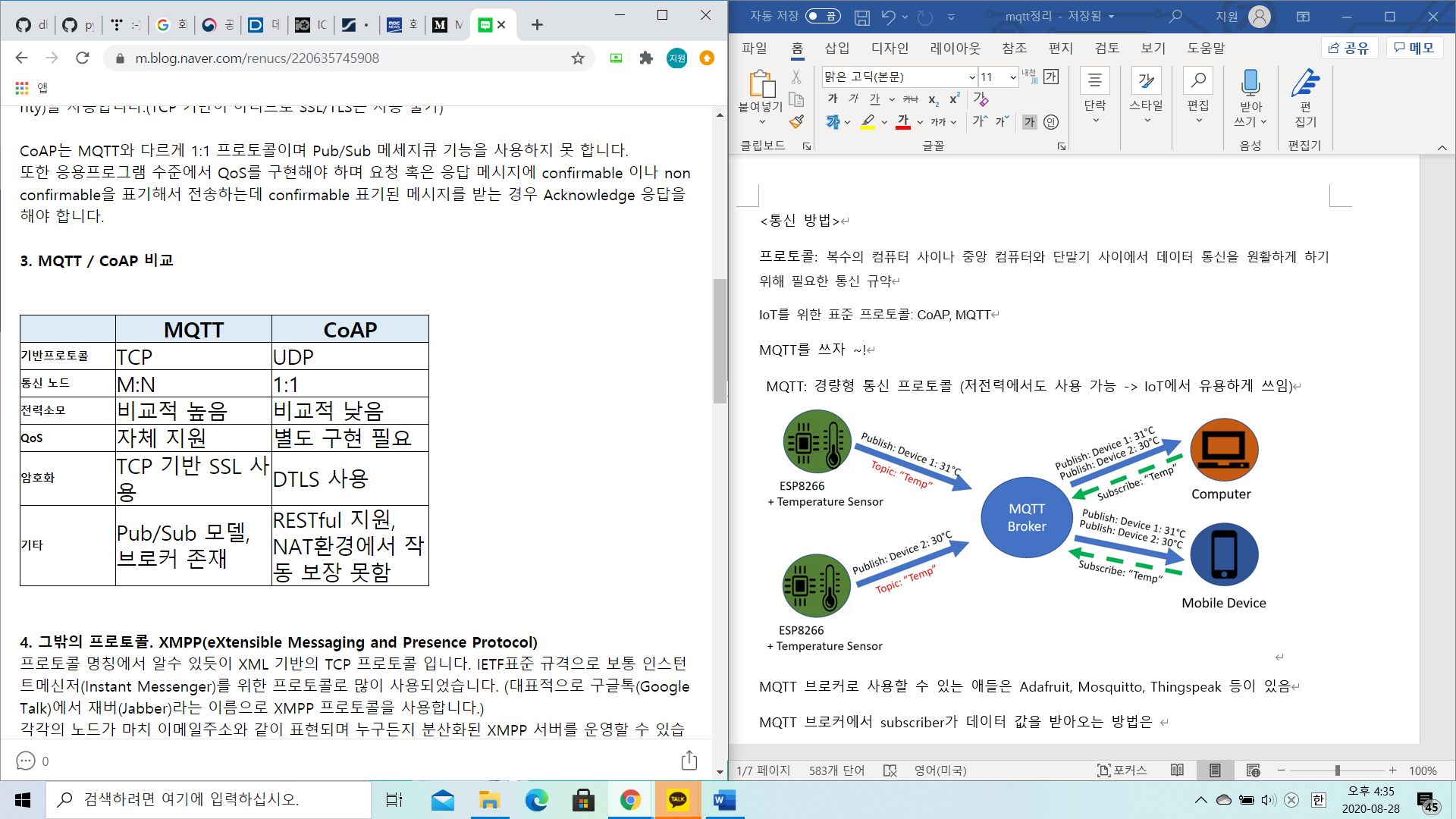
<통신 방법>

프로토콜: 복수의 컴퓨터 사이나 중앙 컴퓨터와 단말기 사이에서 데이터 통신을 원활하게 하기 위해 필요한 통신 규약 -> 프로토콜이 있어야 데이터 통신이 가능하다

IoT를 위한 표준 프로토콜: CoAP, MQTT (경량형이라서)

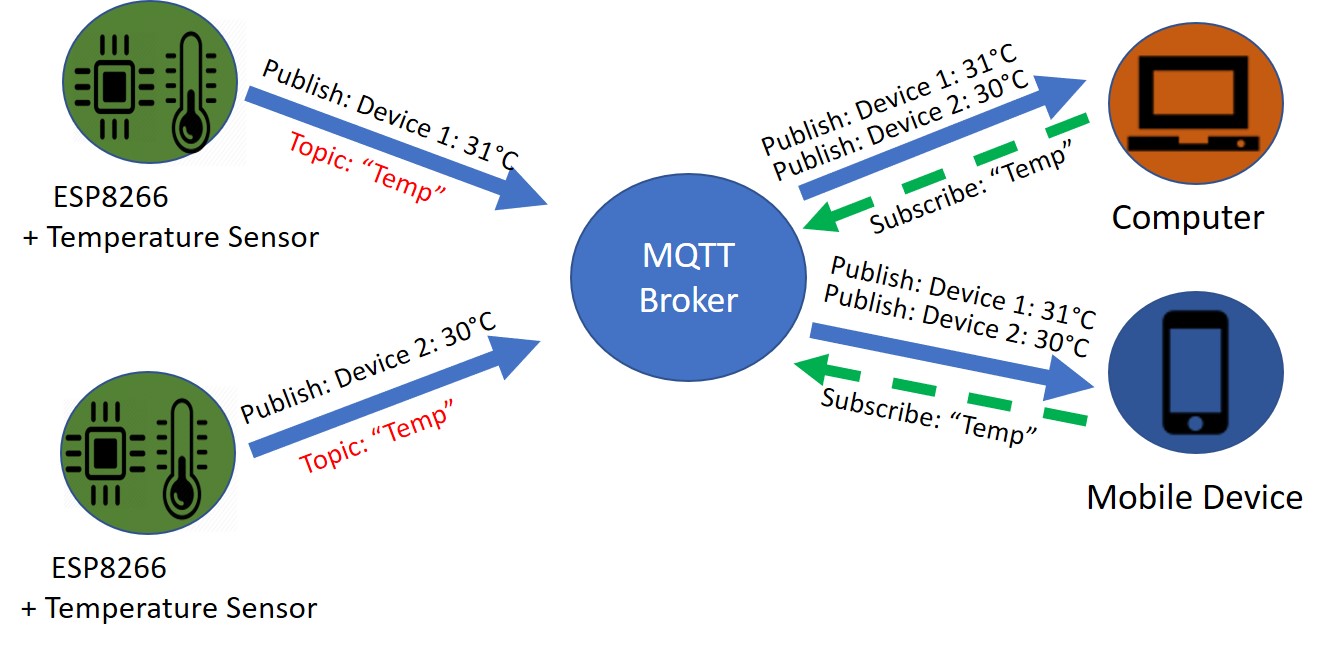


<MQTT를 쓰자>

MQTT: 경량형 통신 프로토콜 (저전력에서도 사용 가능 -> IoT에서 유용하게 쓰임)

MQTT는 Broker, Publishing and Subscribing, Topics, Quality of Service만 알면 된다! !

* 모든 디바이스나 클라이언트들은 Broker(서버)를 통해 커뮤니케이트한다.



클라이언트는 subscriber가 될 수도, publisher가 될 수도 있다. 특정 Topic을 구독해서 broker를 통해 subscribe해 데이터를 받아오거나 특정 Topic에 대한 메시지를 broker로 publish 해 준다. 모든 연결이 broker를 통해 이뤄지기 때문에 한 클라이언트가 항상 직접적으로 다른 클라이언트들과 연결되어 있지 않아도 된다.

Topic이란 클라이언트에게 보내질 메시지들의 종류를 categorize하는 방법이다. 클라이언트는 특정 토픽에 대한 메시지를 publish할 수도, subscribe할 수도 있다. 토픽은 계층적 카테고리로 이루어져서 여러 토픽에 대한 메시지를 받고 싶다면 wild card를 쓰면 된다.

Broker는 이 모든 메시지들을 받아서 해당 토픽을 subscribe하는 모든 클라이언트들에게 메시지를 전달해준다.

Quality Of Service: 서비스 질 보장 레벨으로 무선 통신망으로써 갖는 통신의 불안정함을 보장해주기 위해 제공하는 서비스로 세가지 종류가 있다.

* 0: 보내고 보낸 거에 대한 feedback이 없음 -> 손실 가능성이 있음, lightest
* 1: 브로커에게서 확인 메시지 받을 때까지 계속 보냄. 최소 한 번은 가게하지만 중복 메시지를 받을 수도 있음.
* 2: 각 메시지가 단 한 번만 가도록 보장해줌.

그래서 얘를 쓰기 위해 개발되어져 있는 cloud MQTT broker 서비스 플랫폼을 쓰면 된다.

* HiveMQ(free), mosquito(free), Adafruit IO, Thingspeak 등

처음에는 Thingspeak를 사용하려했는데, Thingspeak는 파이썬이랑 ESP8266 라이브러리를 지원하지 않고, 무료 버전의 경우 15초에 한 번씩만 값을 받아오기 때문에 적절하지 않다고 판단했다.

Adafruit IO가 ESP8266와 파이썬 라이브러리, MQTT, 대시보드를 전부 지원하기 때문에 유료이더라도 얘를 사용하는 것이 괜찮을 듯하다. (Blynk는 MQTT를 지원하지 않음.) 대신 얘도 무료 버전은 필드 수 제한이 있기 때문에 유료 버전 구매하는 게 좋을 것 같다. Adafruit IO는 1초에 한 번씩 데이터 값을 받아올 수 있다.

저번 시간에 막혔던 부분: 같은 종류의 센서들이 각각 서로 다른 NodeMCU에 연결되어 한 피드(채널)에 publish 한다면 브로커나 다른 클라이언트에서 publisher들을 구분할 수 있는 방법.

* 1. IP주소 받아오기 – 와이파이를 사용하는 경우 끊어지고 재접속하게 되면 바뀔 가능성이 있다.
* 2. node마다 각각의 피드(채널)쓰기

한 노드당 하나의 채널(피드)을 만들고, 해당 채널명을 노드 번호로 지정한다. (채널 하나에 여러 노드를 넣으면 동기화 문제가 발생할 수 있다,,~)

한 채널에는 각각 세가지 종류의 필드를 만든다.

* 필드1: 연기센서 값
* 필드2: 온도센서 값
* 필드3: LED

이 때 4방향으로 나누어지는 LED 방향을 쉽게 구분하기 위해 4 방향에 각각 1, 2, 4, 8의 값을 부여한다. (ex – 동 : 4 서: 1 남 : 8 북 : 2)

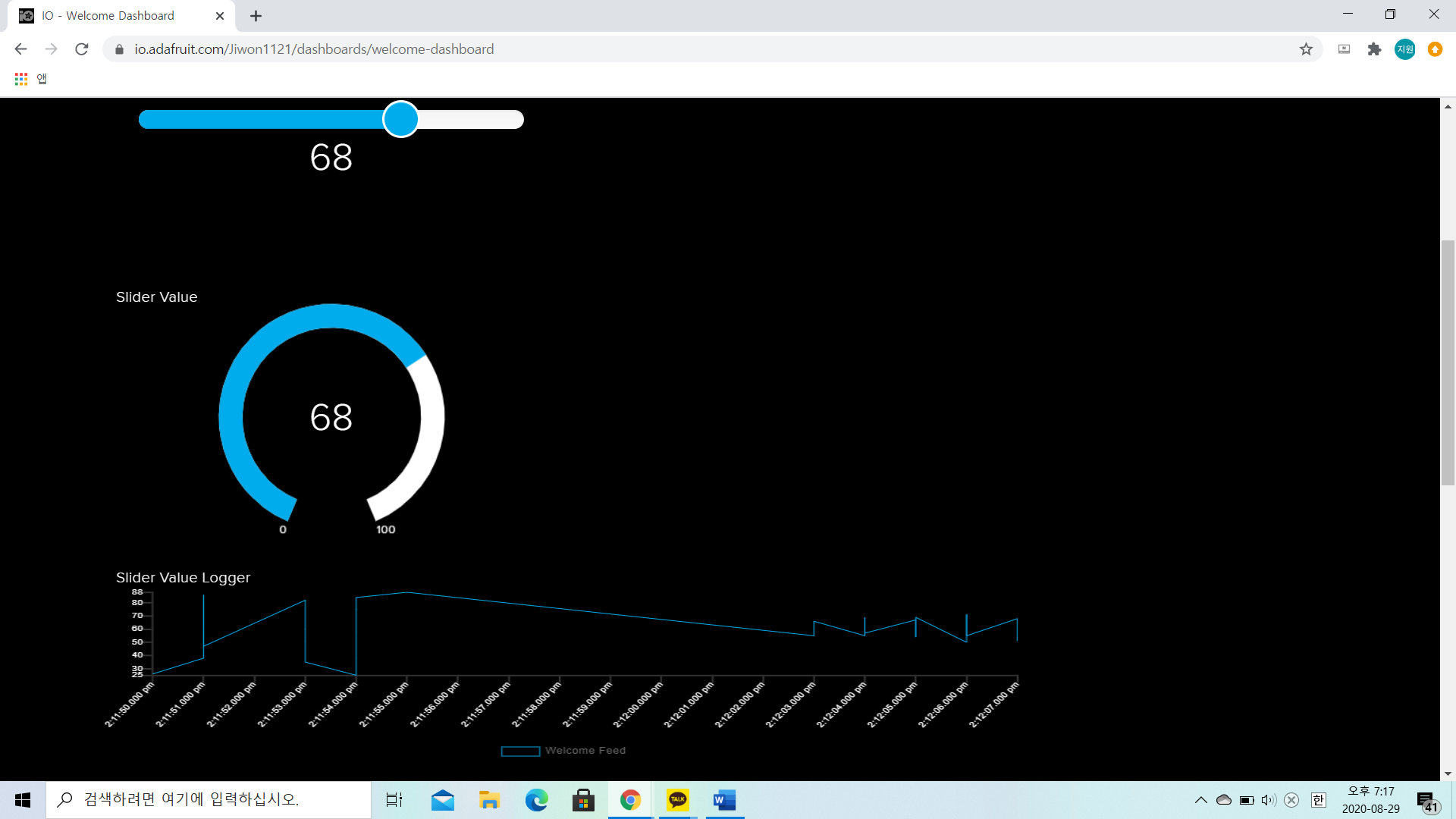
그리고 led에 값을 전달할 때 방향을 더한 값을 전달하면

더한 값만 보고도 어느 방향에 불을 키면 되는지 쉽게 파악할 수 있다.

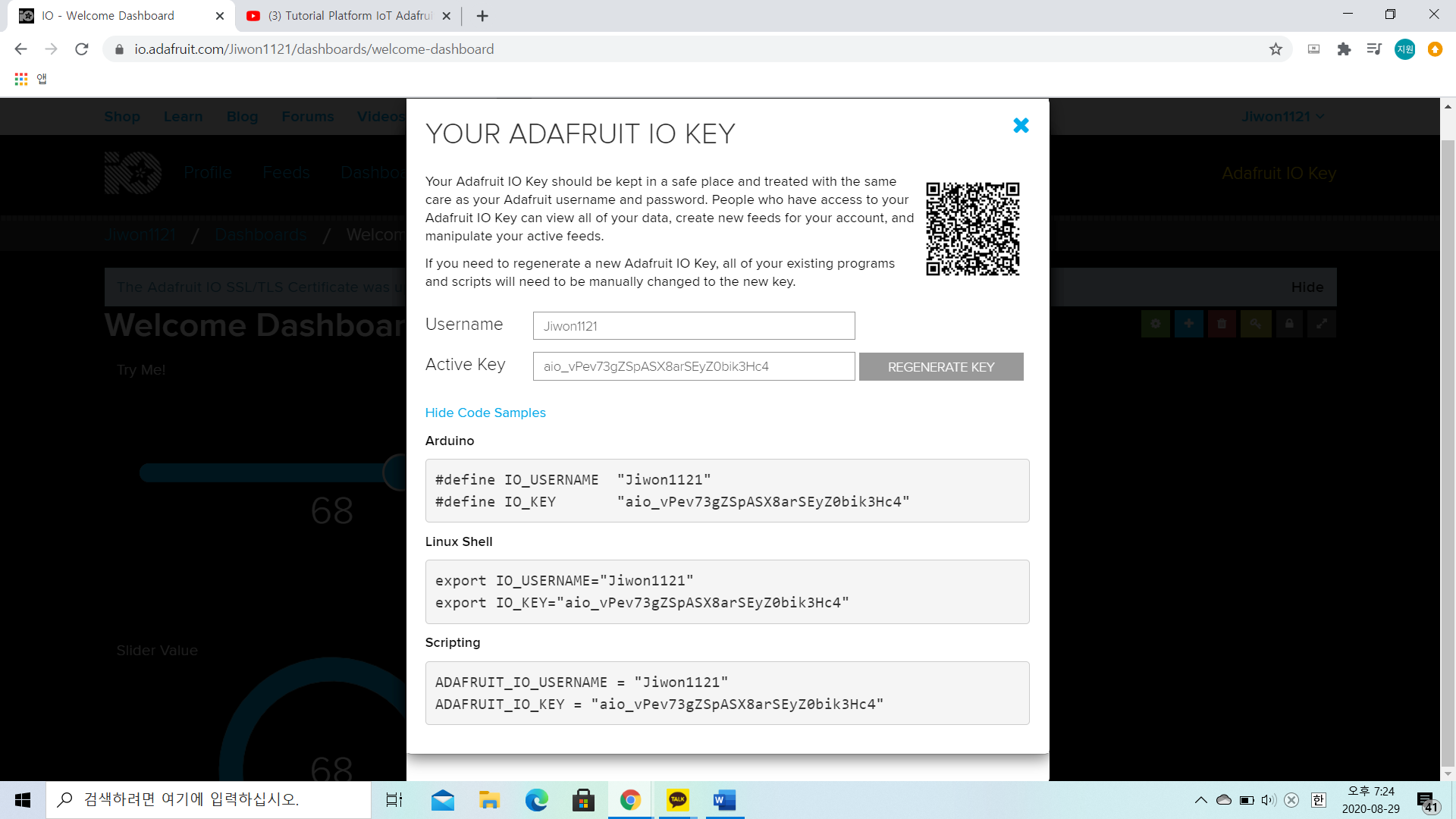
Ex) 동, 서에 불을 켜야할 경우 5를 전달하면 이진수 변환만 해서 불을 켜주면 됨.

\*위, 아래 방향까지 표시해야한다면 16, 32로 늘리면 된다!

(Adafruit IO 대시보드&key)



데이터 값이 들어오면 대시보드에 이렇게 띄워준다.



Adafruit에서 제공하는 개인별 key를 가지고 데이터를 주고받을 수 있다.

<NodeMCU에서 MQTT 브로커로 데이터 값 publish하기>

1. 아두이노 IDE 설치

ESP8266보드는 아두이노 기본 IDE에 포함되지 않아 수동으로 보드 리스트에 추가해줘야 한다.

1. 속성에 아두이노 보드 관리자 url에 <http://arduino.esp8266.com/stable/package_esp8266com_index.json>
2. Tools>Board>Board Manager – esp8266 설치
3. Tools>Board – NodeMCU 1.0 보드 선택

특정 아두이노 보드를 연결하면 정상적으로 포트인식이 되지 않는데 이유는 표준 시리얼 to USB 칩을 사용하지 않고 별도의 칩을 사용하기 때문이다. 그래서 운영체제에 따라 별도의 드라이버를 설치해주어야 아두이노 보드와 컴퓨터간 정상적으로 시리얼 포트를 인식할 수 있다.

1. 포트 선택(usb케이블 필요)
2. 속도 115200 디폴트
3. Pc 연결 : <https://github.com/nodemcu/nodemcu-devkit/tree/master/Drivers> - CH340x 드라이버 설치 (노드엠씨유 usb 드라이버 – pc 연결 설치) / 만약에 이 사이트에서 안되면 httpsL//goo.gl/mzDCTF 여기서!
4. 리부팅 – 드라이브 재인식

<알고리즘 짤 때 > - 브로커에서 값 받아오고 거기로 led 방향 값 보내는 건 paho 라이브러리를 쓰면 될 거라고 생각 중ㅎㅎ 근데 안 될까봐 걱정

루프 {

센서 값 받아오기

If(불이 나면)

위험도 업데이트

LED 키라고 하기

}

* 충돌 문제??

\*\*\*\*\*\*\* 잘 모르겠는 거ㅠㅠ

거의 모든 ESP8266 모듈들은 공장 출하시 UART serial IO 속도가 115200으로 정해져 있다.

Arduino Mega 와 같이 HW Serial 이 두개면 문제가 없지만, Arduino Uno/Nano의 경우, 하나밖에 없는 HW Serial을 USB 연결용으로 사용해 버리므로 문제가 된다.

결국, Arduino Uno/Nano는 ESP8266 와 SoftwareSerial로 연결되어야 하나, SoftwareSerial 은 115200처럼 높은 baud rate를 지원하지 않는다.

그래서 연결하려는 arduino 가 Uno/Nano 라면 BAUD RATE를 변경해 줄 필요가 있다.

이거 때문에 firmware 업데이트 같은 걸 해줘야 되는데 여기서 잘 안돼서 막힘(ESP-01)

NodeMCU 중에서 일부는 firmware 업그레이드를 해야되는 것이 있고 안해도 되는 것이 있다고 하는데 기준을 잘 모르겠다. 만약 펌웨어 탑재가 필요한 NodeMCU는 <http://www.14core.com/how-to-use-nodemcu-flasher-step-by-step-tutorial/> 이 페이지를 참고하면 쉽게 할 수 있다고 한다.